Eye Gaze Detection by CNN model

Lau Kin Fung, David (3035274959)   Mok Tsz Chung, Tony (3035278694)

Introduction

Eye gaze shows the direction that a person is looking at. From everyday conversation to psychological research, eye gaze is an important index showing the attention level of human. In the following, we explore how to make use of a machine learning approach to detection the eye gaze, which is represented by 7 directions including upper left, upper right, left, right, lower left, lower right and centre. After that, we try to use the machine learning model to detect some eye disease such as esotropia and hypotropia.

Motivation

Due to the fast growth of technology, eye gaze estimation can be used in different kinds of applications such as gaming using VR and MR. In fact, eye gaze estimation not can be applied to gaming research but also can be used in the psychological research. For example, when a driver is driving a car, the camera installed in the car can detect the attention level of the driver by detecting the direction of eye gaze so this can ensure the safety in the road. Moreover, eye gaze can be used to detect what the person is thinking in his mind. The following picture captured from George's paper[2] showing when the person is looking at different directions, each direction represents a psychological meaning. As a result, eye gaze detection becomes more useful and important in different categories, which raises our interest to explore more in eye gaze detection and how to use it to predict the eye diseases.

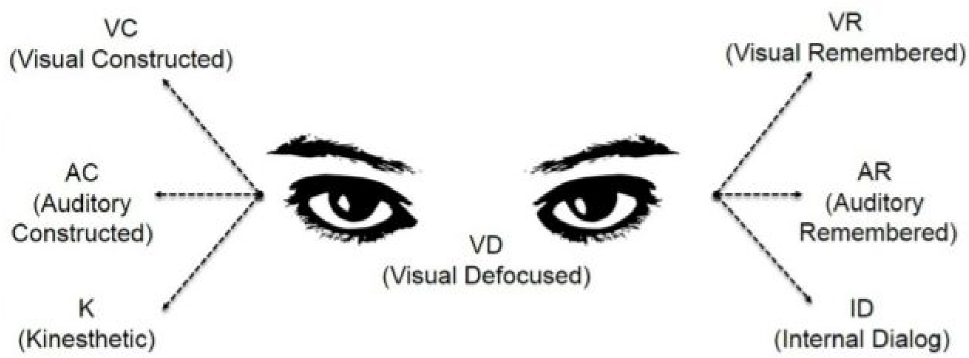

Fig. 1: Eye Accessing Cues (EACs)

Innovation Point

The innovation point of the project is the detection of eye disease (Strabismus) as everyone can use their webcams installed on the smartphones or computers to check whether they suffer from strabismus or not. By using our machine learning model, people can check this type of disease at their home first before they consult the doctors and the speed of the checking process is really fast, which is within 1 to 2 seconds.

Related work

There are 2 major existing methods to detect eye gaze[5][6]. The first one is user calibration-free system[11]. As a pair of the camera is used to estimate the eye parameters, it does not require the user to calibrate, which is more user-friendly. However, the high complexity of the system due to 2 cameras and 5 illuminators used causes the increase in price. The second major method to detect eye gaze requires user calibration. The system is called Calibration using static markers[12]. This system detects gaze by detecting the edge of iris and centre of iris. It is a user calibration required system as it takes few seconds to do calibration process and gets the coordinate on the screen. Thus, both systems are not good for MR and AR application due to high cost and time-consuming calibration process. Our algorithm is used to overcome disadvantages mentioned.

Proposed method (Modified from George's Paper[2])

Step 1: Face detection

First, we extract the face from the original image using Haar Feature-based Cascade Classifiers in the OpenCV library[3] and the sample result is shown in Fig. 2. After extracting the face, we resize the face to 96x96 and convert the RGB image into grayscale for the landmark detection in Step 2.

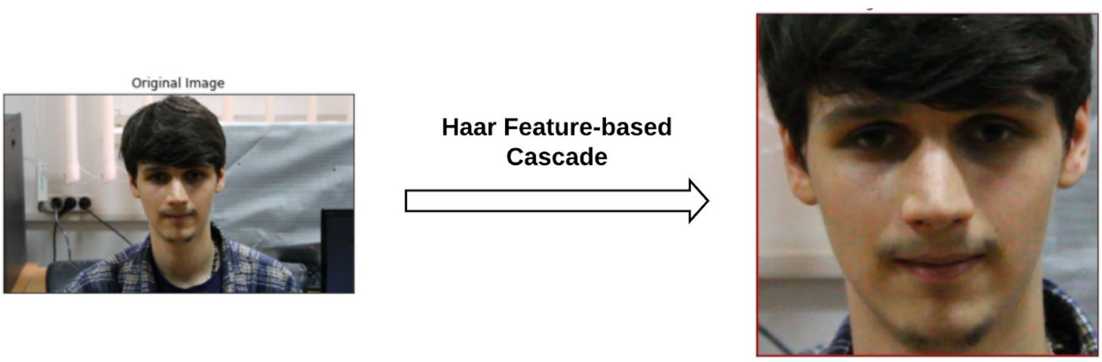

Fig. 2: Result of face detection before resizing and grayscaling

Step 2: Eye detection by landmark detection

Originally, we try to use OpenCV library to crop out the eyes from the face image. However, it is not accurate enough as the library always misdetects the eyes. Fig. 3 shows an example of misdetection.

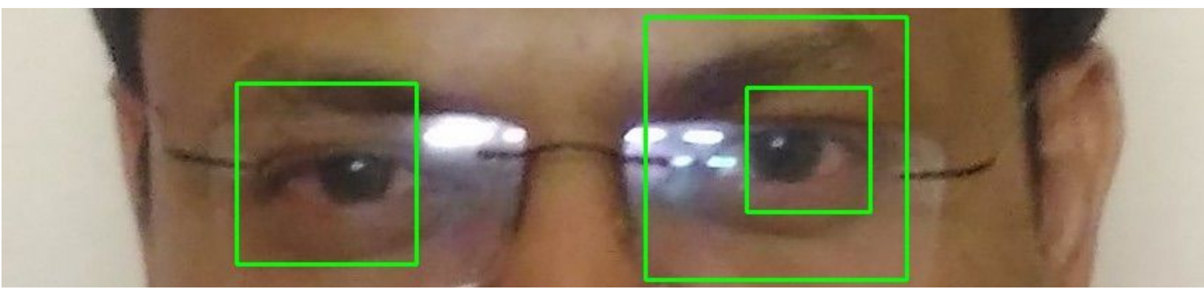

Fig. 3: Misdetection of eyes using OpenCV libaray

(1) Landmark detection: Thus, we build a CNN model to capture the face landmark location, which has 15 points. For the eye part, the model 10 points and their corresponding names are shown in Fig. 4. The CNN model consists of 12 layers, which are shown in Fig. 6. The first layer is a convolution layer with 16 filters and a 3x3 kernel. The second layer is a max-pooling layer having a 2x2 kernel. The third layer is a convolution layer with 32 filters and a 3x3 kernel. The fourth layer is a max-pooling layer having a 2x2 kernel. The fifth layer is a convolution layer with 64 filters and 3x3 kernel. The sixth layer is a max-pooling layer with a 2x2 kernel. The seventh layer is convolution layer with 128 filters and 3x3 kernel. The eighth layer is a max-pooling layer with a 2x2 kernel. After all the convolution layers and max-pooling layers, we use a flattening layer to flatten the input and a dropout layer to dropout some data points in order to prevent overfitting. Finally, we add a fully connected layer with 30 units (all the landmark coordinates) to output the prediction. The activation function used in the convolution layers is ReLU while the activation function used in the fully connected layer is a linear function.

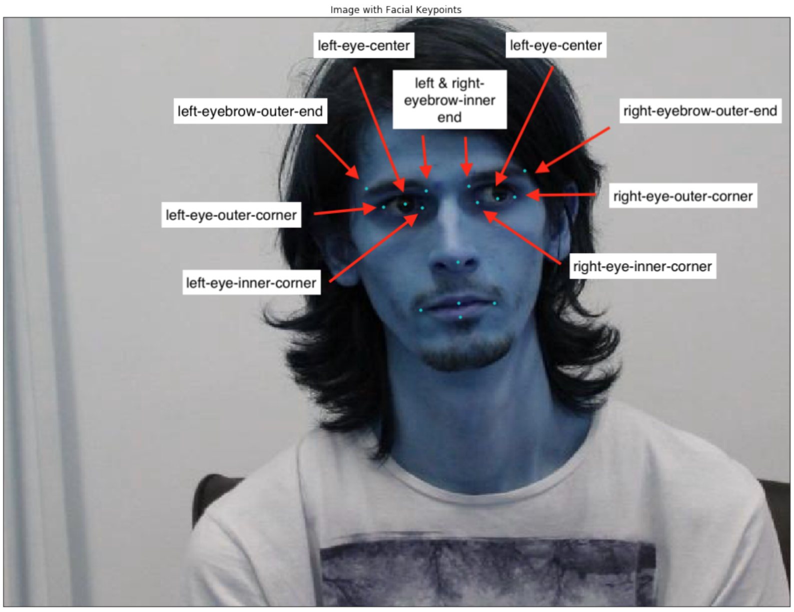

Fig. 4: The names of the landmark points related to the eye

Fig. 5: Sample outputs of landmark detection

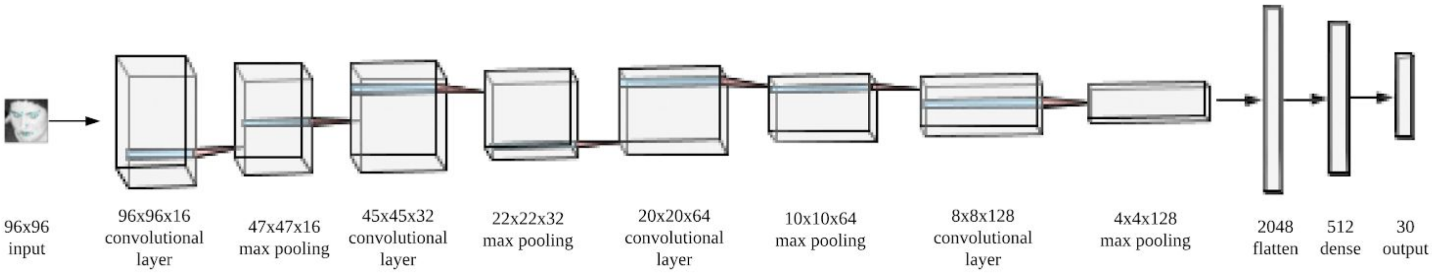

Fig. 6: CNN model for landmark detection(ReLU is used as activation functions)

(2) Eye detection: Then, the landmarks are used to find the location of the eyes. As an illustration, when finding the bounding box of the left eye, we first find out the max(left_eyebrow_inner_end_x - left_eyebrow_outer_end_x, left_eye_inner_corner_x - left_eye_outer_corner_x) and max(left_eyebrow_inner_end_y - left_eye_inner_corner_y, left_eyebrow_outer_end_y - left_eye_outer_corner_y)*2, which are the width and height of the bounding box respectively. Therefore, we can figure out the coordinates of the bounding box of the eyes from the face image extracted in Step 1.

(3) Image cropping and normalization: After getting the coordinates of eyes, we have to map the coordinates back to the original image as the size of the face image is 96x96 so the resolution of the cropped eyes is low. To solve this problem, we need to crop the eyes out from the original image by mapping the coordinates of the face image back to the original image, which has a higher resolution. After that, we need to resize the eyes' images to 42x50 for the eye gaze detection in step 3.

Mapping function:
let the x and y coordinates of a eye in the 96x96 face image be a and b respectively.
let the x and y coordinates of the face in the original image be c and d respectively.
let the width and height of the face in the original image be w and h respectively.
The original x coordinate of a eye in the original image = (a*w/96) + c
The original y coordinate of a eye in the original image = (b*h/96) + d

By using the above mapping function, we can have a higher resolution image of the eyes. The overview of eye extraction is shown in Fig. 7.

Fig. 7: Overview of eye extraction

Step 3: Eye gaze direction classification for one eye

After cropping out the eyes, we start doing the gaze direction classification. We try to modify the CNN classification model by George[2] to classify the eye gaze direction, which is shown in Fig. 8. We aim to do an accurate classification to different classes of “The Eye-Chimera" database”. The CNN model consists of 8 layers. The first layer is a convolution layer with 24 filters and 7x7 kernel. The second layer is a max pooling layer with 2x2 kernel. The third layer is a convolution layer with 24 filters and 5x5 kernel. The fourth layer is a max pooling layer with 2x2 kernel. The fifth layer is a convolution layer with 24 filters and 3x3 kernel. The sixth layer is a max pooling layer with 2x2 kernel. The seventh layer is a flatten layer in order to flatten the input. The eighth layer is a fully connected layer with 7 units to output the eye gaze direction. The activation function used in the convolution layers is ReLU while that used in the fully connected layer is softmax. We use the left eye and right eye datasets to train two separate CNN model.

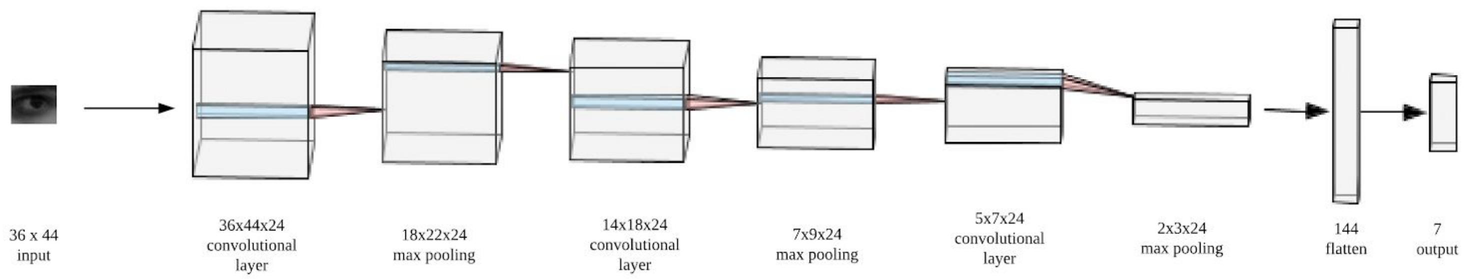

Fig. 8: CNN model for eye gaze direction classification

Step 4: Eye gaze detection

To detect the direction of both eyes, we need to combine the result of the left eye and right eye CNN model, i.e. the probability of each class. Then, we add the probabilities together and divide them by two. Finally, we choose the class having the highest probability.

Dataset

For face landmark detection, we make use of “Facial Keypoints Detection dataset”[4]. There are 2283 images for training and 1782 for testing. Each image has size 96x96 in grayscale with landmark coordinates.

For gaze direction classification, “The Eye-Chimera database” is used. There are different images of human, with each looking at different directions. The distribution of images with different labels is listed as follow:

Class label	Number of images
Center	325

DownLeft	133
DownRight	126
Left	135
Right	138
UpLeft	136
UpRight	142

Experiment

1. Implementation

We implement the experiment using python3 in Jupyter Notebook. The image processing is implemented by openCV[3], numpy[7] and PIL Library. The model is built on Tensorflow backend using Keras[8]. The data splitting and evaluation is operated using Scikit learn Library[9].

2. Evaluation procedure

To evaluate the performance of the model, we split the dataset into two completely disjoint subsets (80%/20%). We had also used data augmentation to enhance the performance of the model. Two CNN model is used to train each eye separately. When we test the performance, we combine the result from both classifiers to obtain the class label of the test images.

3. Result

(1) *Confusion matrix analysis*: Comparing the confusion matrix of the following 3 cases:

1. classify using left eye only (Fig. 9)
2. classify using right eye only (Fig. 10)
3. combining 2 classifiers (Fig. 11)

We found that 3 to 4 class labels with precision higher than 90% if only one classifier is used. After combining the two classifiers, 6 labels are having precision higher than 90%, which shows the improvement of overall prediction.

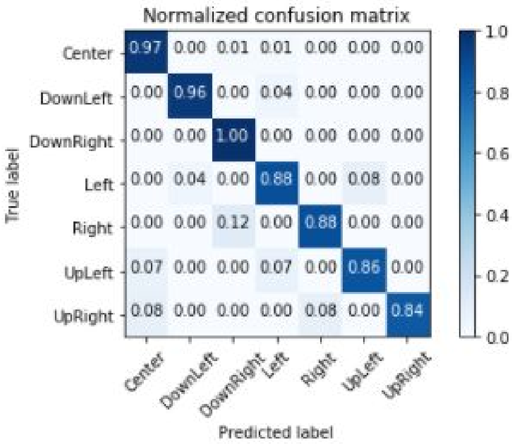

Fig. 9: Confusion matrix of left eye only

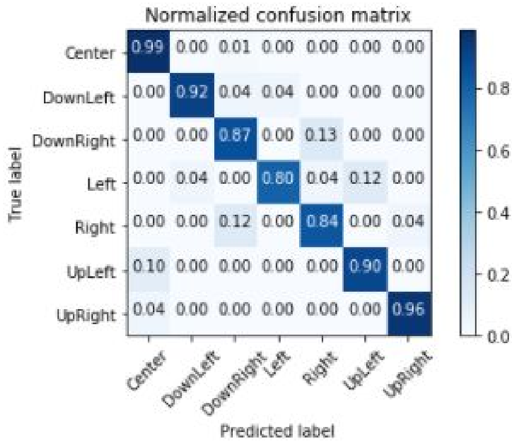

Fig. 10: Confusion matrix of right eye only

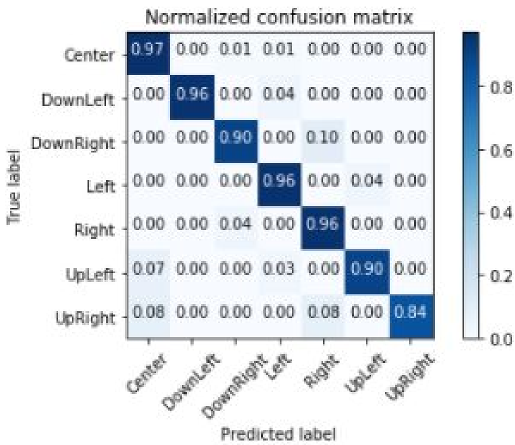

Fig. 11: Confusion matrix of combining classifiers

(2) *Precision and recall:* After 5 batches of testing, the result of different indicators is shown in the table below: the precision and recall are high in both batches and the classifier is reliable due to small standard deviation value.

Indicator	**Mean**	**Standard deviation**
precision	93.6%	6.18%
recall	92.7%	4.89%
F1 score	92.9%	2.99%

4. Performance

The following table shows the specification of the computer used and the mean of classification time. It is fast enough for a normal computer.

Operation System	Windows 8.1
Processor	Intel(R) Core(TM) i7-6700 CPU @ 3.40GHz

average classification time	0.128s

<u>Extension: Strabismus detection</u>

<u>1. Background and classification</u>

Strabismus(squint) is a condition that patients cannot look in the same direction[14]. According to statistic, 4% population suffers Strabismus[13].

<u>2. Classification</u>

Fig.12 shows the conditions of different types of Strabismus. Thus, the above classifier can extend its usage to detect different types of Strabismus.

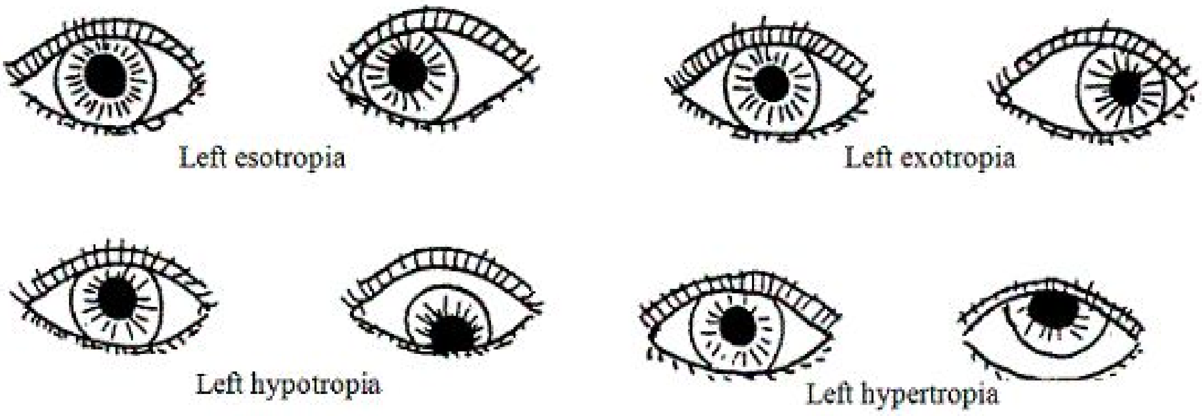

Fig. 12: Types of strabismus[14]

Thus, our classifier can be modified as follow to classify Strabismus:

Type	Left eye class	Right eye class
Normal	Center	Center
Left esotropia	Right	Any
Left exotropia	Left	Any
Left hypotropia	DownLeft/DownRight	Any
Left hypertropia	UpLeft/UpRight	Any
Right esotropia	Any	Right
Right exotropia	Any	Left
Right hypotropia	Any	DownLeft/DownRight
Right hypertropia	Any	UpLeft/UpRight

Proposed method (Modified from the eye gaze detection)

To detect different types of Strabismus, we use a similar approach for eye gaze detection. For the final prediction step, we need not combine the result of two CNN model. Instead, we find out the classes predicted by two CNN model and the corresponding direction. As a result, the eye directions of right and left eye are known and we can conclude the type of Strabismus.

ACKNOWLEDGEMENTS

Thank Dr. Corneliu Florea for “The Eye-Chimera” database.

References

[1] R. Vranceanu, C. Florea, L. Florea, C. Vertan "NLP EAC Recognition by Component Separation in the Eye Region", 15th International Conference on Computer Analysis of Images and Patterns CAIP 2013, York, UK, 27-29 Aug. 2013, LNCS vol. 8048, Eds. Wilson, R.; Hancock, E.; Bors, A.; Smith, W., pp. 225-232, Springer Verlag, Berlin Heidelberg 2013, ISSN 0302-9743, ISBN 978-3-642-40246-3

[2] GEORGE, Anjith; ROUTRAY, Aurobinda. Real-time eye gaze direction classification using convolutional neural network. In: Signal Processing and Communications (SPCOM), 2016 International Conference on. IEEE, 2016. p. 1-5.

[3] Doxygen, “OpenCV documentation”, 2018, Retrieved from: https://docs.opencv.org/3.4.1/index.html

[4] Kaggle, Facial Keypoints Detection, 2017, Retrieved from: https://www.kaggle.com/c/facial-keypoints-detection/data

[5] KIAT, Lim Choon; RANGANATH, Surendra. One-time calibration eye gaze detection system. In: Image Processing, 2004. ICIP'04. 2004 International Conference on. IEEE, 2004. p. 873-876.

[6] GWON, Su Yeong, et al. Estimation of Gaze Detection Accuracy Using the Calibration Information-Based Fuzzy System. Sensors, 2016, 16.1: 60.

[7] SciPy.org, Numpy and Scipy Documentation, Retrieved from: https://docs.scipy.org/doc/

[8] Keras.io, Keras Documentation, Retrieved from: https://keras.io

[9] Scikit learn Documentation, Retrieved from: http://scikit-learn.org/stable/documentation.html

[10] Github, Facial Keypoints Detection, 2017, Retrieved from: https://github.com/jamesrequa/Facial-Keypoint-Detection

[11] Model, D.; Eizenman, M. User-calibration-free remote eye-gaze tracking system with extended tracking range. In Proceedings of the 24th Canadian Conference on Electrical and Computer Engineering, Niagara Falls, ON, Canada, 8–11 May 2011; pp. 1268–1271.

[12] Zhu, J.; Yang, J. Subpixel eye gaze tracking. In Proceedings of IEEE International Conference on Automatic Face and Gesture Recognition, Washington, DC, USA, 20–21 May 2002; pp. 124–129.

[13] MERDLER, Ilan, et al. Strabismus and mental disorders among Israeli adolescents. Journal of American Association for Pediatric Ophthalmology and Strabismus {JAAPOS}, 2017, 21.3: 185-189.

[14] Department of Health, HKSAR. Health Information: Eyes / Vision, retrieved from: https://www.studenthealth.gov.hk/english/health/health_ev/health_ev_squ.html

Distribution of work

David: ML modeling and implementation(Landmark detection and gaze classification)

Tony: background research, data preprocessing, report