Eyetracker Based on Image Recognition Technology and YOLOv8 Implementation

Zonghan Li

School of Internet of Things, Xi'an Jiaotong-Liverpool University, Suzhou, China

Zonghan.Li2103@student.xjtlu.edu.cn

Abstract. Since most of today's eye trackers are based on infrared reflections which means the eye tracker needs a camera, it also needs an extra infrared light gadget. This paper refers to the shortcomings of modern infrared eye-tracking devices. And then design an experiment to collect eye-tracking datasets using only a camera and implement an eye-tracking system deployed on a head-mounted human-computer interaction device by using image recognition techniques and key-point detection techniques. This method makes the eye tracker more complicated. This paper designs an experiment by getting the eyeball data from different people in different environment backgrounds to create a new eye-tracking dataset for head-mounted interactive devices and transform that dataset into a '.txt' format file usable by YOLO. Then use this dataset to finish the transfer learning which is based on the YOLOv8, which trained a theoretically realisable model of an eye-tracker. Thereby, the efficiency of the usage of the eye-tracker is optimized in principle, allowing people to use the eye-tracker without relying on an infrared device, but only a camera.

Keywords: Eyetracker; Image recognition; YOLOv8.

1. Introduction

With the development of modern computer science technology and the continuous improvement of people's requirements for quality of life. Head-mounted human-computer interaction (HCI) devices such as Virtual Reality (VR) devices and Augmented Reality devices (AR) are gradually starting to appear in real life. This phenomenon is because traditional HCI devices usually require people to operate them or they are usually not portable. This makes the usage efficiency of traditional devices extremely limited in lots of scenarios. For example in some operations in extreme environments such as geological exploration, mining, disaster relief, etc. And since the head-mounted device is easier to operate (no need for cumbersome touch operations) and portable. This makes it possible to improve the efficiency of usage in some scenarios to a certain extent. For example, the new product Apple Version Pro which was released by Apple on June 6, 2023, has been attempted to improve the status by applying it to the ophthalmology and medicine field [1, 2]. While eye-tracking technology can help make the usage of head-mounted devices more efficient, this technology can capture real-time information about where people's eyes are gazing. Please imagine that: If people can control a head-mounted device simply by looking at it without any extra action, this will undoubtedly improve the efficiency of the device's usage significantly. However, the conventional eye tracker device usually uses a camera to capture an image of the eye and uses an infrared light source to illuminate the eye. The infrared light creates reflective spots on the cornea of the eye. The camera then captures the positional information of these reflective spots and tracks the eye movements by analyzing the positional changes of these spots [3]. This makes the process of the device acquiring information about the position of people's eye gaze somewhat cumbersome. However, obtaining the reflective spots of infrared light can be eliminated if the eye tracker is implemented through image recognition technology. In other words, that is, if eye-tracking were to be implemented using image recognition technology, it would be possible to theoretically implement eye-tracking without the need for an infrared device, but only a camera.

This paper designs an original gadget consisting only of a camera and an eyeglass frame and uses it to simulate a head-mounted human-computer interaction device. Then a simple eye tracker model through YOLOv8 which finished transfer learning. This paper also experimentally demonstrated that

eye-tracking can also be implemented using only a single camera and image recognition technology. It was theoretically demonstrated that the current status of eye-tracking devices, which mostly require an extra infrared light unit, could be improved.

2. Principle Explanation and Experimental Design

2.1. Principle of Experiment

This eye-tracker is designed on the principle of image recognition technology for use on head-mounted human-computer interaction devices (VR or AR devices). It can use the camera to capture a real-time image of the user's eye, and then perform image classification and key point detection on this image to identify the area that the user is currently gazing at, as well as the location of the user's eye. These zones need to be defined in advance. If the object in this research used for the experiment is a screen mounted on smart glasses, the experimenter needs to divide the zones for this screen in advance.

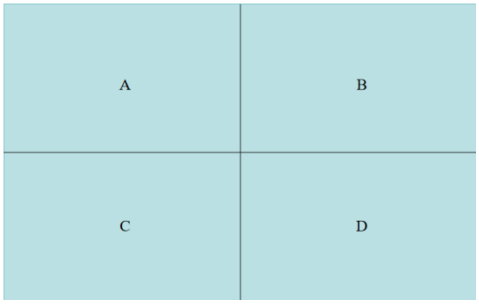

Fig. 1 Divide the screen area (Original)

2.2. Experiment Prepare

As shown in Figure 1, the experimenter needs to divide this screen into 4 areas A, B, C, and D in advance. The screen is divided in this way to explain the principle of the experiment. In reality, it is necessary to divide the screen into more areas, that is, to divide the screen into more areas that cover a smaller area. Because the final target is to be applied to a head-mounted device, the relative position of the screen and the user's eyes should remain the same. And experimenter will need to use a frame to hold the camera so that the relative position of the camera and the frame remains the same. Below is the gadget that the author designed for this experiment.

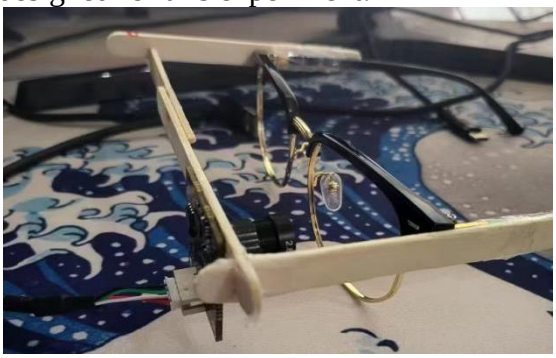

Fig. 2 Headset gadget demo (Original)

According to Liversedge, when a person is reading or observing, both eyes usually turn in synchronous, a movement known as binocular coordination. In the vast majority of studies on reading and observation, only one of the two eyes is usually recorded, so this gadget was only photographed on the left eye [4]. Referring to Yan's research, when collecting eye data, the camera should not directly cover the whole eyeball, otherwise, it will prevent the user from observing the objects directly in front of them [5]. Therefore the author staggered the position of the camera at an angle to the

position of the eyeball to prevent error interference when collecting data. Figure 3 shows the relative position of the camera and the eyeball.

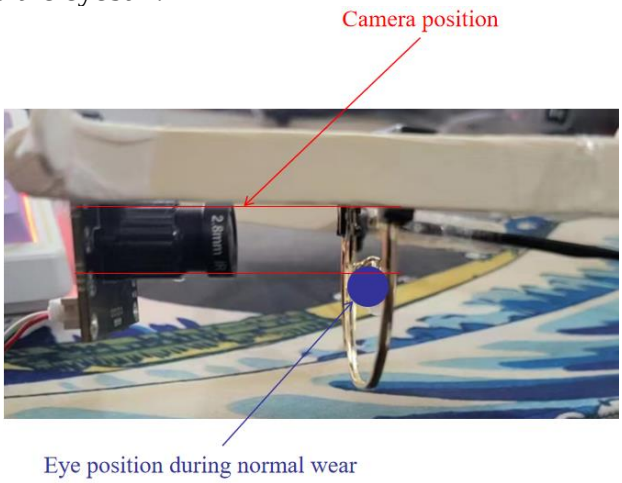

Fig. 3 Explanation of the angle of deviation of the eye from the camera (Original)

2.3. Data Collection

This section references the method of collecting eye movement data in Krafka's study, where they used a method of collecting data by having the user look at points of light on a screen[6].

The author divided the screen into 5 regions, which are the center of the screen, the top left region, the bottom left region, the top right region, and the bottom right region. The regions are divided as shown in the Figure 4:

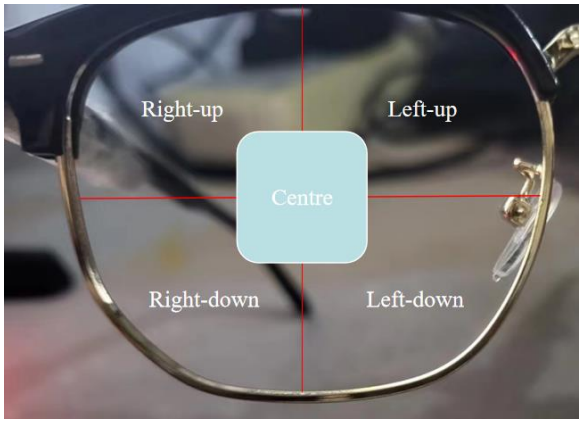

Fig. 4 Region dividing in experiments (Original)

Because these areas are orientated relative to the user, it should be the case that what the readers of the paper are seeing is the result of a mirror image inversion. To facilitate the collection of the dataset, the author labeled each of these areas to make it easier for the user to gaze at the corresponding correct area. Of course, as mentioned above: If this research wants to achieve more accurate eye-tracking to find the exact location of the user's current gaze, then in practice these larger regions should be divided into smaller sub-regions, which of course makes it much more difficult to collect the dataset and train the model. Therefore, this paper will focus on demonstrating the feasibility of implementing eye-tracking through this method. That is, if the model can find where the user is currently gazing by correctly identifying the classes of the five regions, then adding more regions can also be successful, but only consuming more time.

2.4. Dataset Built

The author collected eye movement data from people of different ages, eye sizes, genders, ethnicities, and pupil colors. And used them to build a new dataset and classify it into 5 classes (C means Centre, L_U means Left-up, R_D means Right-down) according to the above classification.

The samples in this dataset are all images with a resolution of 1920x1080. Because at this resolution you can get a clearer picture of the sample, on the other hand, many real-life camera devices support this resolution:

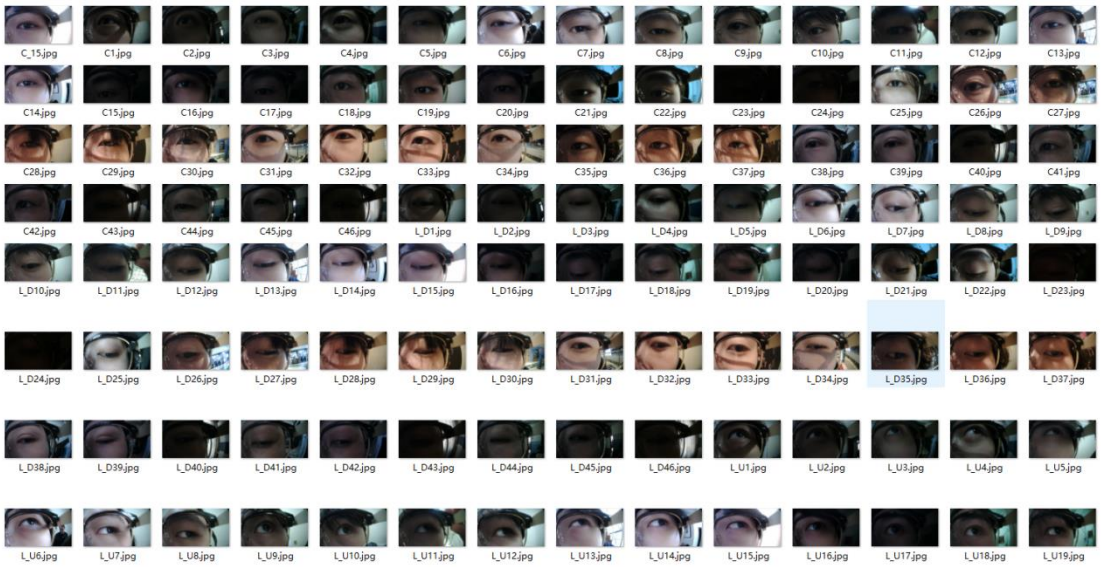

Fig. 5 Data Set Presentation (Original)

As Figure 5 shows. To enhance the robustness of the model so that it can still perform effective recognition in different environments. Therefore the author considered the effect of background when collecting eye movement data: the author collected eye movement data from users with different light intensities and background situations as a training set. On the other hand, when testing the performance of a model, using samples from different background situations as a test set gives a better analysis of the background in which the model has a better performance.

2.5. Training Model Selection

In terms of the model used, the author chose YOLOv8 for transfer learning. This research needs real-time prediction of where the user is looking, which means that recognition speed should be prioritized. YOLO is characterized by its fast processing speed, which allows for fast result prediction. On the other hand, YOLOv8 supports keypoint detection. If the user's eyeball is used as the detection object during training, then the user's eyeball position is predicted and displayed at deployment runtime, which can be used to directly demonstrate the accuracy of the results.

2.6. Dataset Labeling

In this research, the author chooses to label me for dataset labeling. Because this labeling tool supports key point labeling. As above mentioned: sample images should be classified into 3 classes. The label box of each class needs the part that frames the user's eye, but there should be no glass frames in the box. Besides that, this research still needs to label the eyeballs of users as key points. So, the method of labelling should follow Figure 6.

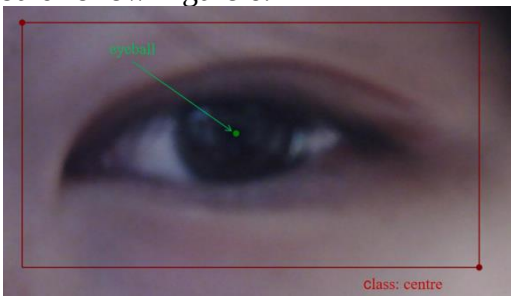

Fig. 6 Labeling situation of the class ‘centre’ (Original)

2.7. Dataset Format Transform

The output file format of the labeling tool Labelme is '.json', but YOLO only could recognize the '.txt' file. So before starting to train the model, the '.json' file must be transformed as '.txt' file. Each '.json' file has one or more label boxes, where the content of each label box can be converted to a single line of text in '.txt' format. The authors summarised the pattern as shown in Figure 7 by interpreting the textual content: The first number represents the class of the object in the current box, which is represented as an integer. For example, when the total number of classes is 5, the first class 'centre' can be converted to the numeric form '0'. The second number and the third are the coordinates of the centre point of the box (x,y). These two values are proportional values indicating the ratio of the coordinates of that centre point to the label box. The fourth and fifth numbers represent the width and height of the label box. They are also proportional values, representing the proportion of the whole image that this label box represents. If that label box contains keypoints, then one or more sets of keypoint coordinates appear after the 5th number. A set of keypoint coordinates contains three numbers, the first and second numbers represent the x, and y coordinates of that keypoint, and the values of these coordinates are also all proportional values. And the third number represents the visibility of that keypoint, i.e., whether or not that keypoint is obscured. If it is obscured it is 1. If it is not obscured it is 0.

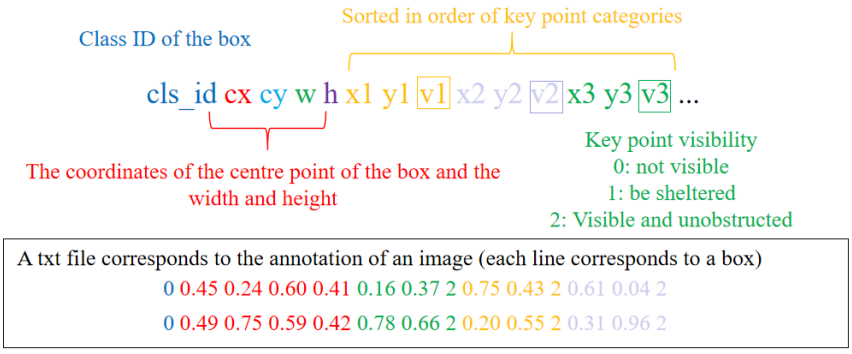

Fig. 7 The method of the '.json' format to the '.txt' format (Original)

3. Model Performance Evaluation and Discussion

3.1. Performance Evaluation

After training the model, the author obtained a simple eye tracker model. It could recognize the region where the user gazing. The model's performance test is in the test set in Figure 8.

Fig. 8 Model performance in the test set (Original)

Besides the test set testing, the authors also tried to test the model in real time by deploying it on the gadget and the results of the test are as Figure 9.

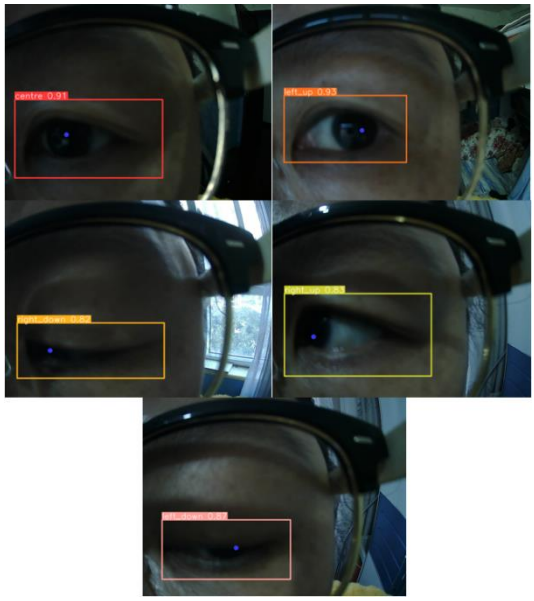

Fig. 9 Real-time test result in gadget (Original)

It can be seen that the confidence of the model's predictions is mostly close to 0.9, which usually means that the model is very confident in its predictions. Since the model outputs a high level of confidence both when run on the test set and when deployed on the gadget for real-time prediction, this means that the model performs well and can recognize images more accurately in a variety of environments.

In addition, YOLOv8 also gives the precision-recall(P-R) curve of the model. Precision is the proportion of samples predicted by the model to be in the positive classes that are truly positive, while recall is the proportion of all samples that are truly in the positive classes that are predicted by the model to be in the positive classes. If a model performs well, it should have both high precision and recall. Therefore, the P-R curve can also be used to evaluate the model performance. It shows in Figure10:

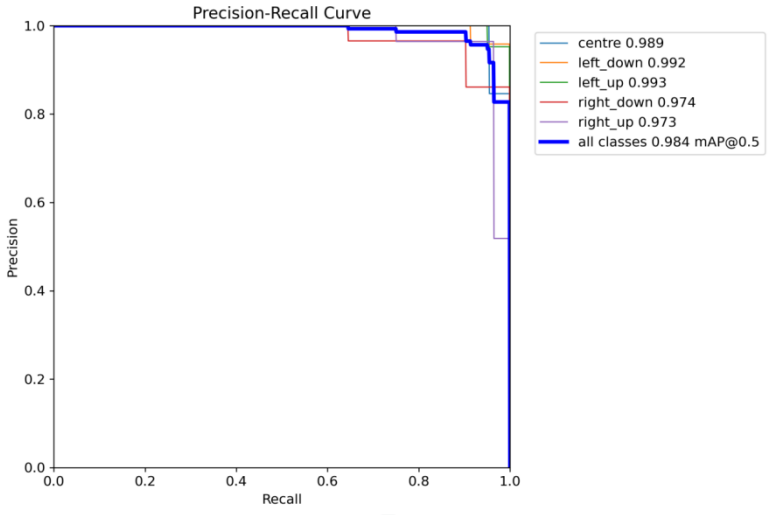

Fig. 10 P-R Curve (Original)

As Figure 10 shows, the P-R curves of this model are all close to the upper right corner, the curves of different classes (centre, left_down, left_up, right_down, right_up) are almost overlapping, and all of them have precision and recall close to 1.0, which suggests that the model is very accurate in recognizing all classes. Figure 10 also mentions "all classes 0.984 mAP@0.5", which means that the model has a mean Average Precision (mAP) of 0.984 at an Intersection over the Union (IoU) threshold of 0.5. It calculates the average precision at different recall rates. The mAP score of 0.984 is very high and indicates that the model performs well.

3.2. Discussion

Therefore, the experiment of implementing a simple eye-tracker using image recognition techniques was very successful. It also shows that it is theoretically possible to improve eye-tracking devices, as most traditional eye-tracking devices need to use an extra infrared device. But with this technology, almost the same effect can be achieved with only one camera, which can greatly reduce the cost of using eye-tracking devices.

However, the technology shown in this paper still needs to be improved in the future. Since the actual situation requires more sub-areas within the visible area to locate the user's gaze more accurately. There is therefore a need to prepare larger datasets, set up more region classes, and use more accurate camera devices to collect samples, which is a future goal of this research.

4. Conclusions

In summary, the authors have combined a single camera and an eyeglass frame to stitch together a simple gadget that can be used to simulate a head-mounted HCL device. The viewing area of this device can be divided into 5 zones, the centre zone, the left up zone, the left down zone, the right up zone, and the right down zone. The authors then composed an eye-tracking dataset by collecting images of users who use this gadget gazing at different areas.

Following this, the authors used YOLOv8 for transfer learning to train a simple eye-tracker model which implemented with image recognition and keypoint detection techniques. In turn, this research proves that it is possible to implement eye-tracking technology with just one camera without the need to install extra infrared devices. Theoretical demonstration of the possibility of improving most infrared eye-tracking devices in reality.

The authors then point out that this technique has some limitations at this stage: if one wants to realise an eye-tracker with a high degree of completion, then the visual area needs to be divided into more sub-areas. In turn, each sub-area needs many samples for training, so this work is very heavy. However, once realised, using an eye-tracker will be easier and less costly because there is no need to add an extra infrared device.

Finally, in the future, the authors will try to collect more samples and try to improve the completion of this technique to make the eye-tracking technique without infrared more complete.

References

[1] Waisberg E., Ong J., Masalkhi M, et al. The future of ophthalmology and vision science with the Apple Vision Pro. Eye, 2023.

[2] Waisberg E, Ong J, Masalkhi M, et al. Apple Vision Pro and why extended reality will revolutionize the future of medicine. Irish Journal of Medical Science, 2023, 1-2.

[3] Holmqvist K, Nyström M, Mulvey F. Eye tracker data quality: What it is and how to measure it. Proceedings of the symposium on eye tracking research and applications, 2012, 45-52.

[4] Liversedge S P, White S J, Findlay J M, et al. Binocular coordination of eye movements during reading. Vision Research, 2006, 46(15): 2363-2374.

[5] Yan Z, Wu Y, Shan Y, et al. A dataset of eye gaze images for calibration-free eye tracking augmented reality headset. Scientific Data, 2022, 9(1): 115.

[6] Krafka K, Khosla A, Kellnhofer P, et al. Eye tracking for everyone. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016, 2176-2184.